Efficient Multi-Exponentiation

Jonathan Bootle

University College London

This document explains a special case of Pippenger's algorithm [Pip80] for efficient multi-exponentiation.

1 Goal

Let $\mathbb{G}$ be a group of prime order $p \approx 2^{\lambda}$. Let $g_0, \dots, g_{N-1}$ be elements of $\mathbb{G}$ and let $e_0, \dots, e_{N-1}$ be elements of $\mathbb{Z}_p$. Assume that $\lambda \geq N$.

Let $G = \prod_{i=0}^{N-1} g_i^{e_i}$.

Problem Given $g_0, \dots, g_{N-1}$, and $e_0, \dots, e_{N-1}$, compute G.

2 Reduction to Multi-Products

We call the case where $e_0, \dots, e_{N-1} \in \{0, 1\}$ a multi-product rather than a multi-exponentiation. The first step will be to reduce the computation of G to the computation of many multi-products.

Set $s \approx \sqrt{\frac{\lambda}{N}}$ and $t \approx \sqrt{\lambda N}$. Let $e_{i,l}$ be the binary digits of e_i.

$$\begin{aligned} G &= \textstyle\prod_{i=0}^{N-1} g_i^{e_i} \\ &= \textstyle\prod_{i=0}^{N-1} \prod_{l=0}^{\lambda-1} g_i^{e_{i,l} 2^l} \\ &= \textstyle\prod_{i=0}^{N-1} \prod_{j=0}^{s-1} \prod_{k=0}^{t-1} g_i^{e_{i,j+sk} 2^{j+sk}} \\ &= \textstyle\prod_{k=0}^{t-1} \left(\prod_{i=0}^{N-1} \prod_{j=0}^{s-1} g_i^{e_{i,j+sk} 2^k} \right)^{2^{sk}} \end{aligned}$$

Set $g'_{i,j} = g_i^{2^j}$ for $0 \leq j \leq s-1$.
Set $e'_{i,j,k} = e_{i,j+sk}$.
Set $G'_k = \prod_{i=0}^{N-1} \prod_{j=0}^{s-1} g_i^{e_{i,sk+j} 2^j}$ for $0 \leq k \leq t-1$.
Then, we have

$$\begin{aligned} G &= \textstyle\prod_{k=0}^{t-1} {G'}_k^{2^{sk}} \\ G'_k &= \textstyle\prod_{i=0}^{N-1} \prod_{j=0}^{s-1} {g'}_{i,j}^{e_{i,j+sk}} \end{aligned}$$

We will now consider the new multi-product problem:

New Problem Given $\{g'_{i,j}\}$, $\{e'_{i,j,k}\}$, compute $\{G'_k\}$.

The new problem has $Ns = t$ input group elements $g'_{i,j}$ and t output group elements G'_k.

2.1 Visualisation

This approach to computing G can be visualised by arranging the binary digits in a matrix.

$$\begin{array}{c|ccccc|c}
 & e_{0,0} & e_{0,s} & e_{0,2s} & \cdots & e_{0,(t-1)s} & g_0 \\
e_0 & e_{0,1} & e_{0,s+1} & e_{0,2s+1} & \cdots & e_{0,(t-1)s+1} & g_0^2 \\
 & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 & e_{0,s-1} & e_{0,2s-1} & e_{0,3s-1} & \cdots & e_{0,\lambda-1} & g_0^{2^{s-1}} \\
\cline{2-6}
 & e_{1,0} & e_{1,s} & e_{1,2s} & \cdots & e_{1,(t-1)s} & g_1 \\
e_1 & e_{1,1} & e_{1,s+1} & e_{1,2s+1} & \cdots & e_{1,(t-1)s+1} & g_1^2 \\
 & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 & e_{1,s-1} & e_{1,2s-1} & e_{1,3s-1} & \cdots & e_{1,\lambda-1} & g_1^{2^{s-1}} \\
\cline{2-6}
\vdots & \vdots & \vdots & \vdots & \vdots\;\;\vdots & & \vdots \\
\cline{2-6}
e_{N-1} & e_{N-1,0} & e_{N-1,s} & e_{N-1,2s} & \cdots & e_{N-1,(t-1)s} & g_{N-1} \\
 & e_{N-1,1} & e_{N-1,s+1} & e_{N-1,2s+1} & \cdots & e_{N-1,(t-1)s+1} & g_{N-1}^2 \\
 & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 & e_{N-1,s-1} & e_{N-1,2s-1} & e_{N-1,3s-1} & \cdots & e_{N-1,\lambda-1} & g_{N-1}^{2^{s-1}} \\
 & G'_0 & G'_1 & G'_2 & \cdots & G'_{t-1} &
\end{array}$$

The input values for the new problem are shown to the right of the matrix in the same row as the binary digits that they correspond to. The output values are shown below the matrix in the same column as the binary digits that they correspond to.

Computing the multi-exponentiation of the inputs with a column of the matrix gives the output below that column.

2.2 Efficiency

The simplest method of computing the new inputs ${g'}_{i,j}$ is using s squarings of g_i, for each $0 \leq i \leq N-1$, which gives a cost of $\sqrt{\lambda N}$ group operations.

3 Computing the Multi-Products

The new problem has the same number of inputs and outputs, so we relabel to simplify notation. Set $M = \sqrt{\lambda N} = sN = t$.

Problem Given $\{g'_i\}_{i=0}^{M-1}$, $\{e'_{i,j}\}_{i,j=0}^{M-1}$, compute $G'_j = \prod_{i=0}^{M-1} {g'}_i^{e'_{i,j}}$.

Let b be some parameter to be determined later. We partition the input group elements into sets $S_0, \dots, S_{M/b-1}$, each consisting of at most b elements. Then, for each set S_i, we compute the set T_i, containing all possible multi-products of elements in S_i. For example, if $S_0 = \{g_0, g_1, g_2\}$, then $T_0 = \{g_0, g_1, g_2, g_0g_1, g_0g_2, g_1g_2, g_0g_1g_2\}$.

Now, we use the elements of the T_i to compute the G'_j. Note that in order to compute the G'_i, we only need to use one element from each T_i.

3.1 Visualisation

$$\begin{array}{c|c|c|c}
S_0 & S_1 & \cdots & S_{M/b-1} \\
g'_0 \; g'_1 \; \cdots \; g'_{b-1} & g'_b \; \cdots \; g'_{2b-1} & \cdots & g'_{M-b-1} \; \cdots \quad g'_{M-1}
\end{array}$$

$$\begin{array}{c|c|c|c}
T_0 & T_1 & \cdots & T_{M/b-1} \\
g'_0 \; g'_1 \; \cdots \; \prod_{i=0}^{b-1} g'_i & g'_b \; \cdots \; \prod_{i=b}^{2b-1} g'_i & \cdots & g'_{M-b-1} \; \cdots \quad \prod_{i=M-b-1}^{M-1} g'_i
\end{array}$$

$$G'_0 \qquad\qquad G'_1 \qquad\qquad \cdots \qquad\qquad G'_{M-1}$$

3.2 Efficiency

Given S_i, which contains b elements, we can compute all possible multi-products using 2^b group operations. There are M/b sets S_i, so computing all of the T_i costs at most $2^b M/b$ group operations.

Given all of the T_i, each G'_j uses at most one element from each, so it costs at most M/b group operations. There are M of the G'_j, so computing all of them costs at most M^2/b group operations.

4 Recombining Inputs

Given the outputs of the multi-product step, we can now compute the final output G. Recall that $G = \prod_{k=0}^{t-1} {G'}_k^{2^{sk}}$

This can be done using $st = \lambda$ squarings.

5 Efficiency Analysis

This approach can be used to compute $\prod_{i=0}^{N-1} g_i^{e_i}$ using $\lambda + M + 2^b \frac{M}{b} + \frac{M^2}{b}$, where $M = \sqrt{\lambda N}$.

Set $b = \log M - \log\log M$. This becomes

$$\lambda + M + \frac{M^2}{(\log M - \log\log M)(\log M)} + \frac{M^2}{\log M - \log\log M}$$

which is

$$\lambda + \frac{M^2}{\log M} + o\left(\frac{M^2}{\log M}\right)$$

Since $M = \sqrt{\lambda N}$, we arrive at a cost of

$$\lambda + \frac{2\lambda N}{\log \lambda N} + o\left(\frac{\lambda N}{\log \lambda N}\right)$$

References

Pip80. N. Pippenger. *On the Evaluation of Powers and Monomials. SIAM Journal on Computing*, 9(2):230, 1980.